

МЕТОДЫ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ НА ПОСЛЕДОВАТЕЛЬНОСТИ ИЗОБРАЖЕНИЙ. МОДИФИЦИРОВАННЫЙ АЛГОРИТМ СМЕСИ НОРМАЛЬНЫХ РАСПРЕДЕЛЕНИЙ

Кривенко Сергей Алексеевич¹, Кабылкайрова Эльмира Сагибальдиевна²

¹ Студент;

ГОУ ВПО Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: krivenko_sergey@mail.ru.

² Студент;

ГОУ ВПО Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: kabilkairova_elmira@mail.ru.

Данная работа посвящена задаче обнаружения движущихся объектов на последовательности изображений и методам её решения. Подробно рассмотрены алгоритмы, основанные на вероятностных моделях. Модифицирован алгоритм смеси нормальных распределений. Проведено тестирование данного алгоритма в различных условиях съёмки, выявлены его сильные и слабые стороны.

Ключевые слова: обнаружение движущихся объектов, вероятностная модель, смесь нормальных распределений, охранная система видеонаблюдения.

METHODS FOR DETECTION OF MOVING OBJECTS IN A SEQUENCE OF IMAGES. A MODIFIED ALGORITHM FOR MIXTURE OF NORMAL DISTRIBUTIONS

Krivenko Sergey¹, Kabilkairova Elmira²

¹ Student;

Dubna International University of Nature, Society and Man,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: krivenko_sergey@mail.ru.

² Dubna International University of Nature, Society and Man,

Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: kabilkairova_elmira@mail.ru.

This work is devoted to the problem of detecting moving objects in image sequences and methods for its solution. Discussed in detail the algorithms based on probabilistic models. Algorithm for the mixture of normal distributions was modified. Testing of this algorithm in various shooting conditions, identifies its strengths and weaknesses.

Keywords: detection of moving objects, probability model, mixture of normal distributions, security surveillance system.

Введение

Большинство методов обнаружения и выделения движения на изображении (разностный, корреляционный, частотный методы [1,2,5]) позволяют в той или иной мере решить задачу обнаружения объектов, но очень часто не соответствуют предъявляемым требованиям, либо по скорости (до 30 кадров в се-

кунду), либо по качеству (шумы камеры, мелкие дрожания, неправильно определяемый динамический фон) обработки данных в условиях работы с видеопотоком. Это происходит, потому что при работе в реальных условиях возникают следующие проблемы:

- динамическое изменение заднего плана;
- изменение освещения;
- движущиеся тени;
- высокая скорость обработки данных;
- шум камеры.

Проблемы изменения заднего плана, наличия различных шумов и изменения освещения наводят на мысль о создании модели динамического заднего плана, как в методах, основанных на вероятностных моделях [3,4,5].

1. Частотные алгоритмы обнаружения движущихся объектов

При выделении небольших, неконтрастных объектов хорошо помогают алгоритмы, анализирующие частотную составляющую изображения. К ним относятся алгоритмы основанные на преобразовании Фурье [1] и на вейвлетах [4]. Рассмотрим задачу нахождения оценок движения на основе преобразования Фурье.

Пусть у нас имеется последовательность цифровых изображений:

$$I(x, y, t) \quad (t = \overline{0, K-1}, \quad x = \overline{0, N-1}, \quad y = \overline{0, M-1}).$$

Предположим, что все изображения имеют равномерный фон, с нулевой яркостью, а объект представлен одиночным пикселем единичной яркости, движущимся с постоянной скоростью.

Спроецируем плоскость изображения первого кадра ($t = 0$) на ось x , что эквивалентно суммированию яркостей пикселей по строкам. В результате взвешенная сумма проекций на ось x в любой целочисленный момент времени равна:

$$g_x(t, a_1) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y, t) e^{i2\pi a_1 x \Delta t}, \quad t = 0, 1, \dots, K-1. \quad (1)$$

Аналогично, сумма проекций на ось y равна

$$g_y(t, a_2) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y, t) e^{i2\pi a_2 y \Delta t}, \quad t = 0, 1, \dots, K-1, \quad (2)$$

где a_1 и a_2 – положительные целые числа.

Применяя одномерное преобразование Фурье к функциям (1) и (2), соответственно получаем:

$$G_x(u_1, a_1) = \frac{1}{K} \sum_{t=0}^{K-1} g_x(t, a_1) e^{-i2\pi u_1 t / K}, \quad u_1 = 0, 1, \dots, K-1 \quad (3)$$

и

$$G_y(u_2, a_2) = \frac{1}{K} \sum_{t=0}^{K-1} g_y(t, a_2) e^{-i2\pi u_2 t / K}, \quad u_2 = 0, 1, \dots, K-1. \quad (4)$$

Связь между частотой и скоростью устанавливается следующими соотношениями:

$$u_1 = a_1 v_1 \quad (5)$$

и

$$u_2 = a_2 v_2. \quad (6)$$

В этих формулах скорость измеряется в единицах пикселей за время всей последовательности.

Для нахождения знака составляющей скорости в направлении оси x вычислим выражения:

$$S_{1x} = \frac{d^2 \operatorname{Re}[g_x(t, a_1)]}{dt^2} \Big|_{t=n} \quad (7)$$

и

$$S_{2x} = \frac{d^2 \operatorname{Im}[g_x(t, a_1)]}{dt^2} \Big|_{t=n} . \quad (8)$$

Поскольку g_x изменяется по синусоидальному закону, можно показать, что S_{1x} и S_{2x} будут иметь одинаковые знаки в любой момент времени n , если составляющая скорости v_1 (вдоль оси x) положительна. Наоборот, разные знаки S_{1x} и S_{2x} показывают, что эта компонента отрицательна. Если какая-либо из величин S_{1x} и S_{2x} принимает нулевое значение, то рассматривается следующий ближайший момент времени $t = n \pm \Delta t$. Аналогичным способом определяется и знак составляющей по оси y (для скорости v_2)

Главным недостатком данного алгоритма является необходимость вычисления преобразования Фурье в реальном времени, что делает его непригодным для решения поставленной задачи.

2. Корреляционный алгоритм

Функция взаимной корреляции двух изображений – функция, показывающая степень корреляции (схожести) двух изображений. Обычно она используется для вычисления степени схожести областей на разных кадрах. Эта функция должна давать единственный максимум только для случая двух одинаковых изображений. Очень часто используется нормализованная функция взаимной корреляции. Формула ее вычисления записывается, как показано в (9)

$$\xi(x, y) = \frac{\sum_{i=1}^m \sum_{j=1}^n F(x, y) \times I(x + i, y + j)}{\left(\sum_{i=1}^m \sum_{j=1}^n F^2(x, y) \times \sum_{i=1}^m \sum_{j=1}^n I^2(x + i, y + j) \right)^{1/2}}, \quad (9)$$

где $\xi(x, y)$ – значение нормализованной функции взаимной корреляции прямоугольной области F размерами $m \times n$ с прямоугольной областью таких же размеров, начинающейся в точке (x, y) изображения I .

Максимальным значением функции при полном совпадении первой области с прямоугольником на втором изображении будет единица. Алгоритмы, использующие функцию корреляции, работают следующим образом:

1. находим несколько значений функции, различным образом накладывая на одну область прямоугольник, взятый с другого изображения;
2. из полученных значений выбираем максимальное;
3. если наилучшее значение достаточно близко к единице, значит мы обнаружили именно тот объект, который попал в «коррелирующий» прямоугольник, взятый на втором изображении.

Степень близости значения функции корреляции к единице – параметр, выбирая значение которого, можно добиться оптимального соотношения количества ошибок ложного обнаружения и пропуска объектов.

3. Разностный алгоритм обнаружения движущихся объектов

Имеем последовательность изображений:

$$I_t(x, y) \quad (t = \overline{0, T-1}, x = \overline{0, N-1}, y = \overline{0, M-1}).$$

Для каждого изображения, кроме первого находим разность с предыдущим, т.е. находим последовательность разностей $D_t(x, y) = I_t(x, y) - I_{t-1}(x, y) \quad (t = \overline{1, T-1})$.

Применив пороговую обработку, получим набор масок

$$M_t(x, y) = \begin{cases} 1, & |D_t(x, y)| < \delta; \\ 0, & \text{иначе.} \end{cases} \quad (10)$$



Рис. 1. Пример работы разностного алгоритма:
а) и б) соседние изображения; в) их разность после пороговой обработки

Отметим два вида ошибок такого алгоритма: переднеплановый пиксель был классифицирован как заднеплановый (например, вследствие шума, или из-за того, что цвет переднепланового объекта совпадает с цветом заднего плана) и заднеплановый пиксель был классифицирован как переднеплановый (например, из-за шума). Изменяя параметр δ (порог) и параметры последующей фильтрации можно регулировать чувствительность алгоритма, т.е. уровни этих ошибок. Несомненными достоинствами данного алгоритма является исключительная простота реализации и высокая производительность. Несмотря на эти достоинства, существует ряд проблем, которые делают этот алгоритм практически неприменимым на практике. На рисунке 1 мы видим пример работы алгоритма. После пороговой обработки мы получили объект, совсем не похожий на человека.

4. Нормальное распределение

В данной модели каждый пиксель заднего плана представляют в виде одномерной нормально распределенной случайной величины (гауссиана) т.е.

$$B_t(x, y) = N_t^{xy}(\mu, \delta^2). \quad (11)$$

Для адаптации к медленным изменениям яркости параметры μ и δ^2 изменяются с каждым новым изображением, в соответствии с формулами:

$$\mu_t = (1 - \alpha_1) \cdot \mu_{t-1} + \alpha_1 \cdot c, \quad \delta_t^2 = (1 - \alpha_2) \cdot \delta_{t-1}^2 + \alpha_2 \cdot (c - \mu_{t-1})^2, \quad (12)$$

где c – значение яркости в текущей точке, α_1, α_2 – параметры позволяющие регулировать скорость обучения.

Общая схема работы алгоритма выглядит следующим образом.

Обучение модели заднего плана по первым N кадрам. Для оценки параметров модели μ и δ^2 для каждого пикселя обычно применяется выборочное среднее и среднее квадратичное отклонение значений в данном пикселе, т.е. $\mu = \frac{1}{N} \sum_{i=0}^{N-1} I_i$, $\delta^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (I_i - \mu)^2$, где I_0, \dots, I_{N-1} – значения в пикселе по первым N кадрам.

Далее для каждого следующего изображения и для каждого пикселя выполняются следующие шаги:

$$\frac{|\mu - I|}{\delta} \begin{cases} < \varepsilon, & \text{пиксел } I \text{ сегментируется в задний план} \\ \geq \varepsilon, & \text{пиксел } I \text{ сегментируется в задний план} \end{cases}$$

Проводится коррекция текущих значений μ и δ^2 для каждой точки изображения по формулам (12).

I. Для устранения шума к полученному бинарному изображению обычно применяется медианный фильтр или операции математической морфологии.

Для выбора параметра чувствительности ε фиксируется некоторое число $a \in (0, 1)$. После этого относительно ε решается уравнение $P\left(\frac{|\mu - \xi|}{\delta} < \varepsilon\right) = a$, где $\xi \sim N(\mu, \delta^2)$.

Число a – это вероятность того, что отклонение, вызванное шумом камеры, попадет в указанную ε -окрестность. Для простоты обычно используют значение $\varepsilon = 3$, которому соответствует значение $a \approx 0,98$, что позволяет избежать проблем с шумом камеры.

5. Смесь нормальных распределений

В случае стандартного алгоритма нормального распределения для обработки каждого изображения используются только 2 «слоя» информации: модель фона и текущее изображение. Возникает идея распространить данный алгоритм на большее число информационных «слоев» [3,4].

В данном алгоритме создается модель сцены, используя смесь нормальных распределений. С поступлением каждого нового изображения модель обновляется, после чего происходит классификация пикселей по принадлежности к заднему или переднему плану. Таким образом

$$I_t = \sum_{i=0}^{K-1} \omega_{t,i} * N_t(x, \mu_i, \delta_i^2), \quad (13)$$

где $N(x, \mu, \delta^2) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(x-\mu)^2}{2\delta^2}}$.

Каждому слагаемому в сумме соответствует «процесс» в пикселе изображения, который характеризуется параметрами нормального распределения (математическим ожиданием и дисперсией) и коэффициентом $\omega \in [0; 1]$, который называется весом и является показателем того, насколько часто данный процесс в данном пикселе попадал в поле зрения камеры. Параметр K (максимальное количество нормальных распределений или информационных «слоев») выбирается в соответствии с ресурсами компьютера, обычно берут значения от 3 до 5. Используя и обучая такую модель можно создать алгоритм, который справляется практически со всеми перечисленными выше проблемами.

Общая схема работы алгоритма выглядит следующим образом:

I. На первом изображении последовательности происходит инициализация модели. В каждом пикселе создается один процесс со следующими параметрами:

$$\omega = 1, \mu = c, \delta^2 = \delta_{std}^2,$$

где c – текущее значение яркости в данном пикселе, а δ_{std}^2 – дисперсия по умолчанию.

Все пиксели классифицируются в задний план.

II. Для каждого следующего изображения и для каждого пикселя:

1. Поиск процесса, которому удовлетворяет значение данного пикселя c . Для каждого процесса в модели применяется порог $\frac{|m-c|}{d} < e$. Если текущее значение c для некоторого процесса удовлетворяет порогу, то данный процесс помечается, как текущий и переходим к подпункту 3. Если порог не выполнен ни для одного процесса, то переходим к подпункту 2.
2. Создание нового процесса. Оценки мат. ожидания и дисперсии выбираются следующим образом: $\mu = c, \delta^2 = \delta_{std}^2$. Если количество процессов в модели уже равно k , то ищется процесс с наименьшим весом, его вес не меняется, а остальные параметры приравниваются к параметрам нового процесса, данный процесс помечается, как текущий и переходим к подпункту 5. Если количество процессов в модели еще не достигло максимума, то новый процесс добавляется к списку процессов, его вес приравнивается 0, этот процесс помечается, как текущий и переходим к подпункту 4.
3. Обновление статистики текущего процесса. Оценки математического ожидания и дисперсии обновляются по формулам (12).
4. Обновление весов процессов. Обозначим за $\omega_{i,t-1}$ вес i -го процесса на предыдущем шаге, а за $\omega_{i,t}$ вес i -того процесса на текущем шаге, тогда

$$\omega_{i,t} = \begin{cases} (1 - \alpha_3) * \omega_{i,t-1} + \alpha_3, & i = k; \\ (1 - \alpha_3) * \omega_{i,t-1}, & i \neq k, \end{cases} \quad (14)$$

где α_3 – параметр, отвечающий за скорость изменение веса, а k – индекс текущего процесса.

5. Классификация пикселя. Алгоритм классификации очень прост: применение порога к весу текущего процесса.

$$M_t = \begin{cases} 1, & \omega_{k,t} \geq \omega_p; \\ 0, & \omega_{k,t} < \omega_p, \end{cases} \quad (15)$$

где M_t – бинарная маска, в которой 1 – передний план, 0 – задний план, ω_p – порог из интервала (0,1).

III. Для устранения шума к полученной бинарной маске применяется операция математической морфологии.

Метод, используемый для классификации пикселей (15) очевидно допускает принадлежность к заднему плану сразу нескольких процессов. Благодаря этому алгоритм способен адаптироваться к таким изменениям сцены как дождь или снег. Кроме того, благодаря системе оценки параметров модели изображения, алгоритм адаптируется как к медленным, так и к быстрым изменениям освещения.

6. Модификация алгоритма смеси нормальных распределений

Для улучшения характеристик алгоритма в процессе разработка в него был внесен ряд изменений:

1. При поиске процесса удовлетворяющего условию, при наличии нескольких процессов, удовлетворяющих условию, выбирается процесс с максимальным весом. Это позволяет избежать повышения веса случайно созданных новых процессов.

2. При классификации какого-либо объекта в передний план имеет смысл сглаживать разницу в весах текущего процесса в разных пикселях данного объекта. Это позволяет обеспечить мгновенное “пропадание” всего объекта целиком, при достижении его веса порога.
3. Если в определенный момент времени вся сцена была классифицирована в передний план, то, скорее всего, это было вызвано одной из следующих причин: резкая перемена освещения (например, включение/выключение искусственного освещения); сдвиг камеры; закрытие объектива каким-либо большим объектом.

Понятно, что классификация всего изображения в передний план не несет какой-либо полезной информации. В этом случае полезно резко снизить веса текущих процессов ниже порога. Это позволит, не тратя времени на стандартную адаптацию, свести период «слепоты» программы к минимуму.

7. Программная реализация и тестирование алгоритма

Описанный выше алгоритм был разработан и оформлен в виде программного модуля, тестирование которого, было произведено как с использованием камер видеонаблюдения так и с использованием архивных записей, отобранных в ходе работы в реальных условиях, близких к тем, в которых работает типичная система видеонаблюдения. Тестовые данные были сняты в различных окружающих условиях, с различной частотой кадров и различным разрешением.

Подобранные видеоданные в таблице 1.

Таблица 1. Видеоданные тестового набора

Название	Тип видеоданных	Условия съемки	Объекты в кадре	Рисунок
Двор	768x288x24, 3-4 кадров/сек	Пасмурно, зима	Машины, люди	1 а)
Ворота	320x240x24, 15 кадров/сек	Солнечно, лето	Машины, люди	1 б)
Вход	384x288x24, 25 кадров/сек	Помещение	Люди	1 в)
Дорога	768x288x24, 4-6 кадров/сек	Пасмурно, зима	Люди	1 г)



Рис. 2. Кадры видеоданных тестового набора

8. Оценка качества обнаружения объектов

При оценке качества обнаружения использовались несколько параметров:

- число обнаруженных объектов;
- число ложных объектов.

Число обнаруженных объектов – объекты, которые не слились с фоном и были определены алгоритмом.

Ложные объекты – шумовые помехи различного типа, такие как: природные явления (дождь, ветер, снег, молнии); изменения фона (раскачивание деревьев, ветвей, листьев, травы) тень и т.д.

Таблица 2. Оценка качества обнаружения объектов

Название	Время теста, мин.	Число объектов	Обнаружено объектов	“Ложные” объекты
Двор	15	35	31	5
Ворота	7	17	13	5
Вход	67	92	88	6
Дорога зимой	12	22	20	3

Как видно из таблицы 2 алгоритм лучше работает в помещении, чем на улице. Наличие длинных теней серьезно влияет на качество работы алгоритма. С учетом того что большинство ложных объектов – это не правильно определенные тени, важной задачей для дальнейших исследований является разработка качественного алгоритма подавления теней.

Заключение

Модифицированный алгоритм смеси нормальных распределений обнаружения движущихся объектов на последовательности изображений в режиме реального времени показывает хорошие результаты в различных условиях съемки. Кроме того, алгоритм позволяет вести обработку кадров на достаточно высокой скорости (среднее время обработки кадра равно 30,44 мс), что согласуется с задачей обработки в режиме реального времени.

В результате исследований было написано программное обеспечение, позволяющее в реальном времени обнаруживать и отслеживать движение объектов в широком диапазоне окружающих условий. Проведено тестирование разработанного программного обеспечения в реальных условиях, на различных видеоданных. Выявленные недостатки определяют дальнейшее направление исследований.

На основе данного алгоритма и программного обеспечения можно создать рабочую охранную систему, способную обнаруживать объекты на последовательности кадров (в видеопотоке) и отслеживать выделенные объекты как внутри помещения, так и на открытом пространстве.

Список литературы

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с. ISBN 5-94836-028-8.
2. Методы компьютерной обработки изображений / Под ред. Сойфера В.А.. – М.: Физматлит, 2001. – С. 784.
3. Chris Stauffer, Eric W. Grimson L. Learning Patterns of Activity Using Real-Time Tracking – 2000. [Электронный ресурс]. URL: <http://people.csail.mit.edu/welg/papers/learning2000.pdf> (дата обращения 22.03.2011).
4. Davies D., Palmer P., Mirmehdi M. Detection and Tracking of Very Small Low Contrast Objects . – 1994. [Электронный ресурс]. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.3305&rep=rep1&type=pdf> (дата обращения 25.03.2011).
5. Prati A., Cucchiara R., Grana C. Detecting Moving Objects and their Shadows: an evaluation with the PETS2002 dataset. – 2000. [Электронный ресурс]. URL: <http://library.graphicon.ru/paper/654> (дата обращения 22.03.2011).